LABORATORIO DE CONTROL, Práctica 2: Control con PYTHON

Programa de Ingeniería Electrónica
Facultad de Ingeniería
Universidad de Antioquia
2025-1

Profesor: Alumno: Grupo: Fecha:

1. RESULTADOS DE APRENDIZAJE

1.1 Representar en Python, el modelo de un sistema
1.2 Caracterizar la respuesta temporal de un sistema de control.
1.3 Introducir el concepto de compensación de los SC.
1.4 Estudiar algunos comandos de Python Control, de frecuente uso en control

2. ACTIVIDAD DE APRENDIZAJE

El estudio de los Sistemas de Control, SC, es necesario para el diseño de los controladores que permitan obtener resultados deseados del sistema. La biblioteca de sistemas de control de Python (python-control) es un paquete de Python que implementa operaciones para el análisis y diseño de sistemas de control de retroalimentación., que ofrece múltiples herramientas para tanto el estudio como el diseño de los SC. En la práctica, se hará una primera aproximación a la solución de temas de control con apoyo del programa

3. Parte 1 – Para desarrollar e investigar en la primer parte del laboratorio

El estudiante debe presentarse a la práctica con *una copia limpia de esta guía* imprea o en PDF donde luego creará un cuaderno en Jupyter Notebook (ó Google COLAB) para desarrollar los siguientes puntos:

3.1 Resolver los numerales 5.1, 5.3.4, 5.5.1 y 5.6.1 de esta guía

3.2 Se haga **una descripción breve**, descripción corta y clara, de los comandos de Python: tic, toc, time, tf, ss, zpk, tf2ss, tf2zp, ss2zp, ss2tf, zp2ss, zp2tf, roots, pole, zero, poly, eig, pzmap. dcgain, minreal, subplot, feedback, , tfdata. Se usarán muy frecuentemente estos comandos y es necesario conocerlos y usarlos adecuadamente. Para más información vea https://python-control.readthedocs.io/en/0.10.1/index.html

3.3 Dé respuestas a las siguientes preguntas, investigue al respecto:

- ¿Qué es el polinomio característico de un sistema?
- ¿Qué son los valores propios de un sistema?
- ¿Qué son el tiempo de estabilización y el sobre nivel porcentual de la respuesta temporal de un sistema ante entrada tipo escalón?

4. PRELIMINARES

El sistema mostrado, figura1,

Fig. 1

puede representar cualquier físico: Hidráulico, neumático, térmico, mecánico, eléctrico, etc. El comportamiento dinámico del sistema se describe por su modelo matemático (expresión matemática que describe el comportamiento dinámico del sistema), el cual se puede obtener por métodos analíticos o por métodos experimentales.

Supongamos que experimentamos con el sistema, el cual se encuentra en un punto de operación estable, y que lo sometemos a un cambio en la señal de entrada, *u(t),* que deja como resultado un cambio en el punto de operación del sistema y, como consecuencia de ello, un cambio en la señal de respuesta, *y(t)*, tal como se muestra en la figura 2. En la figura 3 se muestra las mismas señales, pero referidas a los ejes de valor cero.

Después de un procedimiento matemático, con base en las señales experimentales se obtiene para el sistema un modelo en forma de Función de Transferencia, FT, dado por la expresión

$$G(s) = \frac{ke^{-sTm}}{(\tau_1 s+1)(\tau_2 s+1)} = \frac{1.2e^{-39.2s}}{(59.7s+1)(22.3s+1)} \qquad [1]$$

Donde

T_m : Tiempo muerto; k : ganancia del sistema; τ_1 y τ_2: Constantes de tiempo

Este modelo de segundo orden con tiempo muerto, MSOTM, tiene una respuesta ante entrada escalón que se superpone a la del sistema original en la figura 4.
Observe que el tiempo muerto, con valor en el modelo de T_m = 39.2 segundos, desplaza en el tiempo la respuesta del modelo en esa cantidad de tiempo. Observe la diferencia y la similitud entre la respuesta experimental y la respuesta del modelo

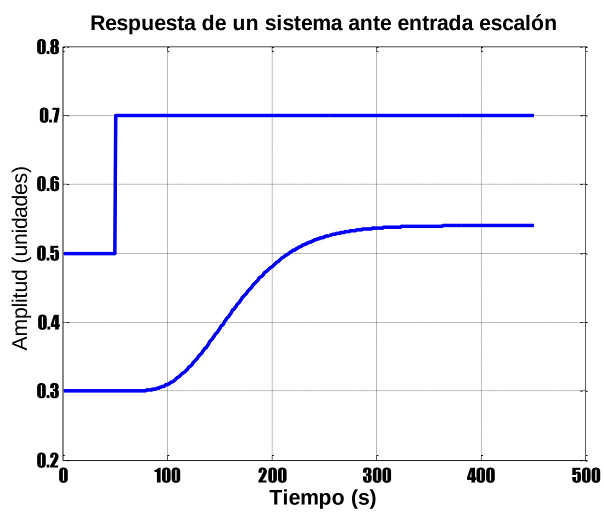

Fig. 2 Respuesta experimental de un sistema físico con condiciones iniciales

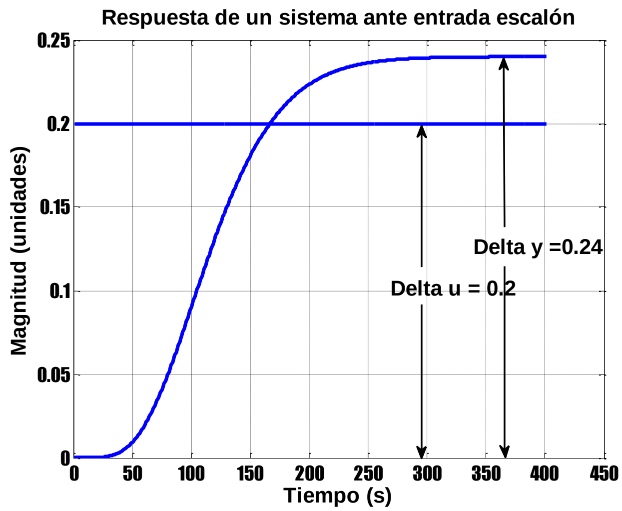

Fig. 3 Respuesta del sistema físico sin condiciones iniciales

5. PROCEDIMIENTO

Nota**:** Envíe a su profesor el archivo ipynb, inmediatamente después de la práctica en el cual vaya dando solución a cada uno de los puntos a desarrollar en el laboratorio.

5.1 Parta del modelo de FT sin tiempo muerto del sistema y halle el **modelo de Espacio de Estado, EE**. Convierta la Función de Transferencia en su correspondiente ecuación diferencial, con coeficiente de la mayor derivada del modelo igual a 1, y use Variables de Estado de Fase, VEF, (tome $x_1(t) = y(t)$ y $x_2(t) = y'(t)$), de modo que la respuesta del sistema es la VEF número 1. Trabaje algebraicamente, sin dar valores a las constantes del modelo. Escriba a continuación:

Modelo de FT sin tiempo muerto:

Modelo de EE sin tiempo muerto:

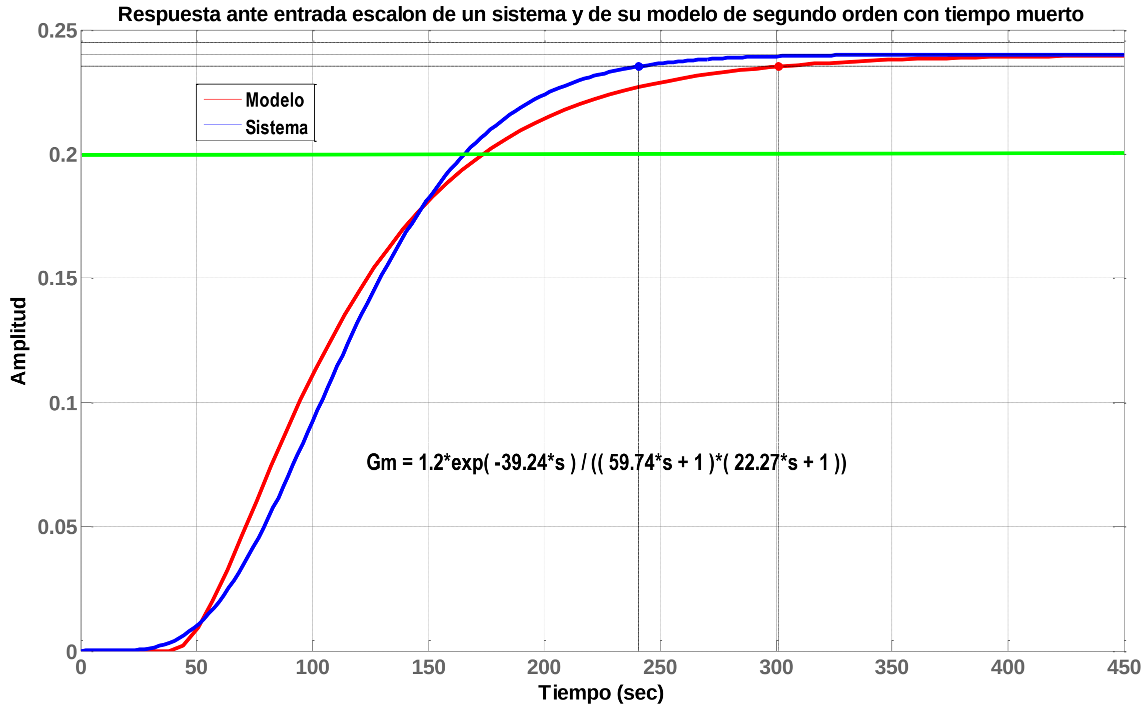

Fig. 4 Respuesta al escalón de un sistema físico y la de su modelo de segundo orden

5.2 Ahora ponga en Python la información de 5.1. Defina en PYTHON las variables ***num*** y ***den*** y ***A, B, C, D***, que representan el numerador y denominador del modelo de FT y las matrices ***A, B, C, D*** del modelo de EE, **sin el término de tiempo muerto.** Use los comandos **tf** y **ss** para crear ambos modelos en ML; llámelos **Gptf** y **Mpss,** .

>> Gptf= tf(num,den) % Numerador y denominador del modelo de FT.
>> Mpss =ss(A,B,C,D) % Matrices A, B, C, D del modelo de EE

Se han definido dos estructuras en el Espacio de Trabajo de Python, ETPython, una de tipo ***f****unción de* ***t****ransferencia*, ***tf*** (*transfer function*), y otra del tipo *espacio de estado*, ***ss*** (*space state*); ellas representan dos modelos diferentes del mismo sistema, por lo que las respuestas de ambos modelos ante entrada escalón deben ser iguales. Compruébelo con los siguientes comandos de PYTHON y ponga título a las gráficas resultantes y a sus ejes.

>> step(0.23*Gptf)
>> figure
>> step(0.23*Gpss)

5.2.1 ¿Cuál es el valor final de las respuestas anteriores? Justifique su respuesta.

5.2.2 ¿Cuáles son los tiempos de estabilización de las respuestas?

Como ayuda para contestar las preguntas anteriores, utilice la función https://python-control.readthedocs.io/en/0.10.0/generated/control.step_info.html.

5.3 Ahora agregue la característica de tiempo muerto a cada modelo. Para ello, primero use el comando *get* para conocer la información que guarda cada estructura. Compare la información que arroja este comando para ambos modelos

5.3.1 Use cualquiera de los siguientes comandos para completar sus modelos con la característica de tiempo muerto

>> Gptfm = Gptf, Mpssm = Mpss % Se crean dos variables a las cuales se les agregará tiempo muerto

>> Gptfm.OutputDelay=39.2 ó Gptfm.InputDelay=39.2

>> Mpssm.OutputDelay=39.2 ó Mpssm.InputDelay=39.2

En Python lea acerca de PADE https://python-control.readthedocs.io/en/0.10.0/generated/control.matlab.pade.html#control.matlab.pade para aproximar el tiempo muerto

Verifique que Gptfm y Mpssm ya incluyen el tiempo muerto

5.3.2 Ejecute el comando *figure* para crear una nueva ventana de gráficas, y con el comando *subplot* grafique verticalmente la respuesta de los modelos con tiempo muerto ante entrada escalón de valor 0.23. Haga que la respuesta de Gptfm quede en la ventana 1 de subplot y la de Mpssm en la ventana 2. Comente los resultados y no cierre la gráfica.

5.3.3 Ya vimos que las estructuras tipo *tf* y *ss* guardan información que se puede conocer con el comando *get;* los campos de cada estructura ya están definidos y lo que cambia son sus datos. A veces se requiere leer o modificar la información que contiene una estructura y el proceso se hace mediante un programa y no manualmente. Se pide escribir a continuación los comandos a usar para dejar en el ETPYTHON los valores de la variables num, den, a, b, c, d y tiempo muerto que guardan los campos de las estructuras con tiempo muerto. Use letras en mayúscula para las variables en el ETPython, como NUM=num, DEN=den, A=a, etc.

5.3.4 Las estructuras creadas permiten obtener otro tipo de información útil. Use comando apropiados (los que conoció cuando hizo su pre informe) para extraer de las estructuras (con o sin tiempo muerto) los siguientes datos, según corresponda. Escriba a continuación los comandos y los resultados hallados

Ceros del sistema:

Polos del sistema:

Valores propios del sistema:

Polinomio característico del sistema:

Ganancia DC del sistema (ganancia a frecuencia cero)

5.4 PYTHON tiene comandos para convertir o crear estructuras a partir de una estructura existente. Defina los sistemas

```
>> systf = tf(Mpss)       % se define un sistema como TF a partir de una representación de EE
>> sysss = ss(Gptf)       % se define un sistema en el EE a partir de una TF
>> syszpk=zpk(Gptf)       % se define un sistema con factores de ceros, polos y una ganancia a partir de una tf
```

¿Existen otras opciones de conversión de estructuras?

¿systf y sysss son iguales a las expresiones Gptf y Mpss que usted manejó en 5.2?

Comentario:

5.5 Para el sistema de la figura 5, en una nueva ventana , halle la respuesta ***y(t)*** ante la señal ***u(t)*** un escalón unitario; no cierre la figura.

fig. 5

Si se usa el sistema Gptfm(s) para formar un SC No Retroalimentado, SCNR (SC de red abierta), tal como se muestra en la figura 6, Gc(s) se puede considerar el controlador del SC propuesto. En este sistema, ***P(s)*** se concibe como una señal de perturbación; observe que, en el tiempo, ***u(t) = p(t) + m(t)***.

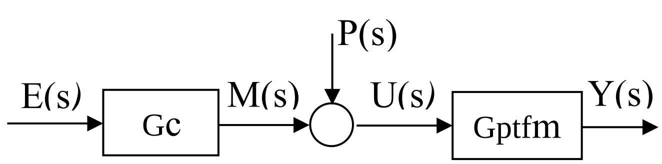

Fig. 6

5.5.1 Con base en la figura 6, escriba a continuación expresiones que relacionen ***Y(s)*** con ***E(s)*** y ***Y(s)*** con ***P(s)*** (simplemente escriba las Funciones de Transferencia correspondientes, FT); llámelas ***Yr(s)*** y ***Yp(s)***, respectivamente; recuerde que el sistema es lineal.

Yr(s) =   ***Yp(s)*** =

5.5.2 Cree en PYTHON una FT para Gc(s) con ganancia 0.3, dos ceros iguales a los dos polos de Gptfm(s) y un polo con constante de tiempo igual al valor del tiempo muerto del sistema. Llame a la anterior FT ***Gctf***, la cual queda en formato o estructura *transfer function, tf*

5.5.3 Cree la variable de red abierta ***Gatfm***= ***Gctf*Gptfm***. Use el comando *minreal* para simplificar la expresión

¿Qué estructura tiene esta nueva variable?:

5.5.4 Si ***E(s)=0*** y ***P(s)*** es el escalón unitario, abra otra ventana de figura y halle, con base en 4.5.1, la respuesta temporal ***yp(t)***.

¿Qué diferencias y similitudes observa de esta respuesta con respecto a la hallada en 4.5?

5.5.5 Si ***P(s)=0*** y ***E(s)*** es el escalón unitario, en una nueva ventana de figura halle ***yr(t)***.

¿Qué diferencias y similitudes observa de esta respuesta con respecto a la hallada en 4.5.Compare tiempos de estabilización, sobre nivel porcentual, valores de estado estable.

Diferencias y similitudes:

¿Considera usted que esta última respuesta es mejor, igual o peor que la inicial?

¿Por qué?

5.6. Considere ahora que todo el sistema de la figura 6 hace parte de un sistema con retroalimentación unitaria negativa, como se muestra en la figura 7

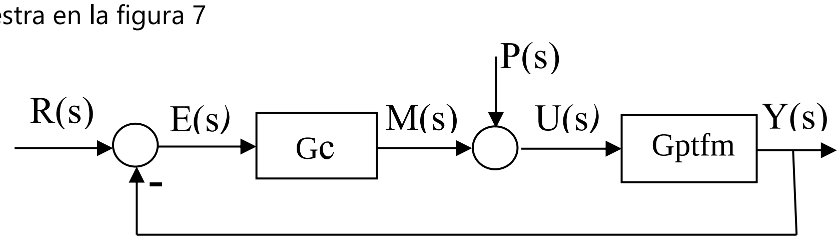

Fig. 7 Sistema reatroalimentado

5.6.1. En algunas versiones de PYTHON se tienen restricciones para manejar sistemas con tiempo muerto en red cerrada y obliga a que se usen modelos de Espacio de Estado, EE. Por lo anterior, se hace necesario convertir *Gptfm(s)* y *Gatfm(s)*, en sus equivalentes de EE. Ejecute los siguientes comandos

```
>> Massm = ss(Gatfm), Mpssm = ss(Gptfm)
```

Nota: Recuerde que una representación en el EE es conceptual y matemáticamente diferente a una representación de FT [1], [2], No obstante, lo anterior, Python permite usar las estructuras de espacio de estado como si fuesen funciones de transferencia, en expresiones de función de transferencia. Esta facilidad permite hallar funciones de red cerrada para sistemas de control con el uso de modelos de EE, de una forma similar a como normalmente se hace con base en funciones de transferencia. Si usa versión más reciente de ML, es posible que pueda hallar FT de red cerrada con FT de red abierta que tengan tiempo muerto. Aclarado lo anterior, se continúa con la práctica

Con el uso de **Massm** y **Mpssm**, escriba una expresión que relacione la señal de respuesta ***Y(s)*** con la señal de referencia ***R(s)*** y ***Y(s)*** con la señal de perturbación P***(s)***. Llámelas ***Gref*** y ***Gper*** respectivamente y créelas en ML.

***Gper* =** ***Gref* =**

5.6.2 Use el comando ***subplot*** y verticalmente grafique las respuestas debidas a la entrada ***R(s)*** y a la perturbación ***P(s)***, cuando cada una de ellas es una señal escalón unitaria. Grafique a continuación las señales obtenidas.

Con respecto a la señal ***P(s)***, ¿Cuál de las respuestas 5.5.4 y 5.6.2 es mejor?

¿Por qué?

¿Se puede decir que la realimentación ha influído de manera evidente en la respuesta ante la misma entrada?

¿Por qué?

Con respecto a la señal ***R(s)***, ¿Cuál de las respuestas 5.5.5 y 5.6.2 es mejor?

¿Por qué?

5.6.3 Halle la respuesta del sistema de la figura 7 cuando el controlador tiene ganancia de 0.2 y compárela con la hallada en 4.6.2. ¿Cuál respuesta es mejor?

6. CONCLUSIONES

7. BIBLIOGRAFÍA

[1] Dorf, R. y Bishop, R., “Sistemas de Control Moderno”, 10ª ed., capítulo 2
[2] Ogata, K., Ingeniería de Control Moderna, 4ª ed., capítulo 4
[3] Nise, N. S. (2020). Control systems engineering. John Wiley & Sons.
[4] Lopez, C. (2014). PYTHON control systems engineering. Apress.